

Article

Not peer-reviewed version

Formulating a Learning Assurance-Based Framework for AI-Based Systems in Aviation

Friedrich Werner , [Johann Maximilian Christensen](#) ^{*,†} , [Thomas Stefani](#) [†] , [Frank Köster](#) , [Elena Hoemann](#) ,
Sven Hallerbach

Posted Date: 11 November 2025

doi: 10.20944/preprints202511.0786.v1

Keywords: AI engineering; W-shaped process; ConOps; OD; ODD; model-based systems engineering; aviation; AI certification; safety-by-design



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Formulating a Learning Assurance-Based Framework for AI-Based Systems in Aviation

Friedrich Werner ¹, Johann Maximilian Christensen ^{2,*}, Thomas Stefani ^{1,†}, Frank Köster ², Elena Hoemann ² and Sven Hallerbach ²

¹ Institute for AI Safety and Security, German Aerospace Center (DLR), Ulm, Germany

² Institute for AI Safety and Security, German Aerospace Center (DLR), Sankt Augustin, Germany

* Correspondence: johann.christensen@dlr.de

† These authors contributed equally to this work.

Abstract

The European Union Aviation Safety Agency (EASA) is developing guidelines to certify AI-based systems in aviation with learning assurance as a key framework. Central to the learning assurance are the definitions of a Concept of Operations, an Operational Domain, and an AI/ML constituent Operational Design Domain (ODD). However, since no further guidance for these concepts is provided to developers, this work introduces a methodology for their definition. Concerning the concepts of the Operational Domain of the overall system and the AI/ML constituent ODD, a tabular definition language for both is introduced. Furthermore, processes are introduced to define the different necessary artifacts. For the specification process of the AI/ML constituent ODD, different preexisting steps were identified and combined, such as the identification of domain-specific concepts for the AI/ML constituent. To validate the methodology, it was applied to the PYCASX system that utilizes neural network-based compression. For the use case, the methodology proved it was able to produce an AI/ML constituent ODD of finer detail compared to other ODDs defined for the same airborne collision avoidance use case. Thus, the proposed novel framework is an important step toward a holistic framework following EASA's guidelines.

Keywords: AI engineering; W-shaped process; ConOps; OD; ODD; model-based systems engineering; aviation; AI certification; safety-by-design

1. Introduction

Artificial Intelligence (AI) and Machine Learning (ML) have become increasingly capable of solving a wide range of tasks and are therefore being adopted by an increasing number of industries in recent years. One of those industries is aviation, an industry governed by one of the strictest certification processes. Still, there are no certification processes available for AI-based systems in aviation, as it has been identified that existing standards, such as ARP4754A, DO-178, or DO-200B, are insufficient for the development of AI-based systems [1–4]. These insufficiencies in the existing standards are caused by the paradigm change from a requirements-driven development to a data-driven development used to develop AI-based systems [5–7]. To solve this issue and create a certification process for those systems, EASA, in cooperation with other stakeholders, is currently developing a guidance framework to ensure the trustworthiness of AI-based systems [5,8]. EASA identified four building blocks necessary to achieve their AI trustworthiness concept and to enable the use of AI-based systems in aviation, namely, the AI trustworthiness analysis, the AI assurance, the Human factors for AI, and the AI safety risk mitigation [8].

The most important component of an AI-based system that distinguishes it from traditional software systems is the AI/ML constituent, containing the ML inference model. For this AI/ML constituent, an Operational Design Domain (ODD) must be defined, capturing the operating conditions

under which the AI/ML constituent must operate as expected [5]. Furthermore, this AI/ML constituent ODD must also provide “a framework for the selection, collection, and preparation of the data” [5], which is used when developing the ML inference model. The AI/ML constituent ODD is therefore crucial, as it is a key component in the developer’s argumentation in the learning assurance process, which is part of the AI assurance. EASA developed concepts such as the AI/ML constituent ODD to allow the Safety-by-Design development of AI-based systems. For these concepts, EASA outlines different objectives that must be achieved by the developer when developing an AI-based system. However, there are currently no regulatory guidelines on the tools and methods that should be used to achieve the various objectives connected to these concepts.

One of the many remaining challenges in developing safe AI-based systems using EASA’s framework, especially the learning assurance process, is how to define the Operational Domain (OD), the AI/ML constituent ODD, and subsequently, the neural networks that are part of the AI/ML constituent. Thus, the goal is to research and apply different Safety-by-Design methodologies to help developers of AI-based systems achieve the objectives of the learning assurance, focusing on the AI/ML constituent ODD [5]. How to achieve the different objectives outlined in the AI trustworthiness analysis and AI assurance is an open question, as EASA does not specify the tools and processes one must use [5]. Therefore, this work contributes a set of processes and formalisms to allow developers to achieve the objectives of EASA’s framework [5] listed in Table 1. Based on these considered objectives, the main focus of this work is on how to define a Concept of Operations, an Operational Domain, and a functional decomposition for the AI-based system, and how the AI/ML constituent Operational Design Domain can be derived based on these artifacts. Additionally, the implications these artifacts have for the AI/ML constituent architecture are investigated. By answering these open questions, a set of processes and tools is introduced that developers can use in the development of an AI-based system to conform to the regulatory guidelines of EASA.

Table 1. The objectives of EASA’s trustworthiness frameworks [5] that are considered for the methodology of this work.

Objective	Description
CO-01	“The applicant should identify the list of end users that are intended to interact with the AI-based system, together with their roles, their responsibilities [...] and expected expertise [...].”
CO-02	“For each end user, the applicant should identify which goals and associated high-level tasks are intended to be performed in interaction with the AI-based system.”
CO-04	“The applicant should define and document the ConOps for the AI-based system, including the task allocation pattern between the end user(s) and the AI-based system. A focus should be put on the definition of the OD and on the capture of specific operational limitations and assumptions.”
CO-06	“The applicant should perform a functional analysis of the system, as well as a functional decomposition and allocation down to the lowest level.”
DA-03	“The applicant should define the set of parameters pertaining to the AI/ML constituent ODD, and trace them to the corresponding parameters pertaining to the OD when applicable.”
DA-06	“The applicant should describe a preliminary AI/ML constituent architecture [...]”
LM-01	“The applicant should describe the ML model architecture.”

This paper describes different processes on how to specify an OD and AI/ML constituent ODD, including necessary formalisms. Section 2 introduces the state-of-the-art on the introduced research

question. In section 3, the developed methodology is presented, and in section 4, the result of the methodology validation is introduced. Section 5 discusses the findings, and section 6 concludes.

2. State of the Art

In aviation, safety-critical systems and components must be certified before they can be deployed. In Europe, this is the responsibility of the European Union Aviation Safety Agency [9]. EASA specifies the regulations that have to be followed and publishes advisories on acceptable means of compliance and guidance material. Based on those regulations, developers can provide evidence that the developed system or component complies with the relevant regulations. The development of acceptable means of compliance and guidance material is supported by the development of industry standards by agencies such as the European Organization for Civil Aviation Equipment (EUROCAE) [10]. As shown in Figure 1, one key framework used in aviation is the ARP4754A/ED-79A standard, which defines the system development process following the V-model. One important input for the V-model is the safety assessment, which has to be conducted according to ARP4761A [11]. For traditional software development, the relevant guidance standard is the DO-178/ED-12 [2], and for hardware development, the DO-254/ED-80 [12] is an accepted means of compliance.

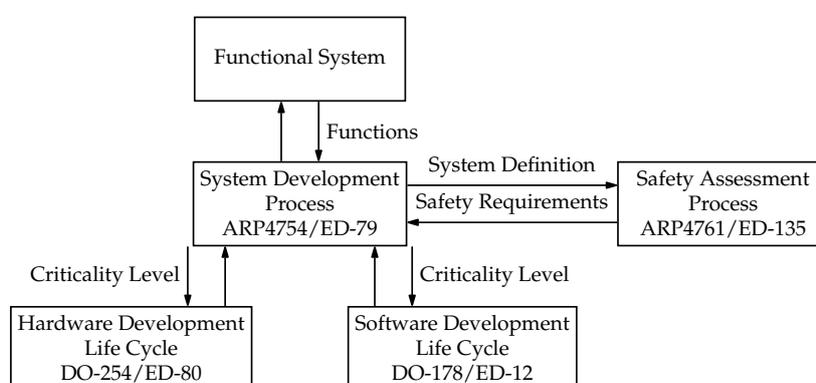


Figure 1. Important standards used in the development of an aviation system to provide evidence to meet the regulatory requirements. Adapted from [13].

However, due to the change to a data-driven development for AI-based systems, the existing standards, together with their corresponding regulatory guidance and rulemaking, are insufficient for the development and certification [4]. To overcome this issue, different regulatory agencies launched initiatives and research projects to solve the identified gaps [8,14–17]. An overview of current advances on different topics, such as the verification of neural networks, can be found in existing literature [13,18]. One important step in that direction is the *EASA Concept Paper: Guidance for Level 1 & 2 machine learning applications* [5]. Here, EASA introduces technical objectives and organization provisions that it sees necessary for the approval of AI-based systems for what EASA defines as level 1 and 2 applications. Furthermore, this concept paper is receiving iterative updates and will be extended to level 3 AI-based systems. EASA defines the AI levels based on the increasing autonomy of the AI-based system. While level 1 offers assistance to humans, level 2 applies to cooperation and collaboration between humans and AI-based systems up to the point of shared responsibilities. Finally, AI-based systems defined as level 3 are autonomous in their decision-making and implementation [8]. Two important aspects introduced in the concept paper are the AI trustworthiness analysis and the AI assurance. The AI trustworthiness analysis contains a characterization and different assessments of the AI-based system. This includes the description of a Concept of Operations (ConOps), which is a refined definition of the AI-based system and the operating environment in which the AI-based system will operate. The description of the operating environment is formalized in the notion of the Operational Domain (OD). The notion to specify the operation conditions for an AI-based system has been developed by the automotive domain [19–25] and has already been adapted by the maritime [26] and railway [27] domains. In the automotive domain, the concept to describe the operation conditions at the system level

is described as the ODD. In contrast, the EASA naming convention uses the term Operational Domain (OD) for a similar purpose. This difference in the naming convention is important to acknowledge, as EASA uses the term ODD at the AI/ML constituent level, where it refers to a different concept. The concept of the AI/ML constituent ODD will be described in more detail in the next paragraph. In the automotive domain, one key property of the specified operating environment is the hierarchical taxonomy-based structure [19,20,28–30]. This structure allows the standardization [19,20] of the conditions that have to be identified, as well as strategies [22,25,31] to identify these conditions for the operating environment. How to specify such an OD for an AI-based system in aviation has been previously explored in different domains such as air traffic management [32,33], airborne collision avoidance [34–38], and unmanned aircraft [39].

With AI assurance, EASA addresses the specific guidance required for the development of an AI-based system [5]. This consists of the development and post-ops AI explainability and as well as the learning assurance. Learning assurance is a new concept introduced by EASA to define a development assurance method that can be applied to the development of an AI/ML constituent. EASA defines learning assurance as all actions required to ensure, with a sufficient level of confidence, that errors have been identified and corrected in the learning process to satisfy the required level of performance for the AI/ML constituent. For the implementation of the learning assurance, EASA proposes its developed W-shaped process [5]. While other works from different domains [7,10,40–42] already introduced processes or frameworks with similar ideas, the introduced W-shaped process adapted and extended those to fit into EASA's framework. Furthermore, additional novel concepts were introduced in the W-shaped process, also by other works [5,43]. One such important novel concept is the AI/ML constituent ODD. With the AI/ML constituent ODD, a developer must define the operating conditions under which the AI/ML constituent is expected to operate as intended [5]. This AI/ML constituent ODD has to include a finer level of detail compared to the operational domain that is defined at the system level to allow the use of the AI/ML constituent ODD in the data and learning management. The specification of an AI/ML constituent ODD is important, as the defined attributes and ranges are used to construct data quality requirements and to verify if the collected dataset for the development of the ML inference model is complete and representative of the expected later operating conditions of the AI/ML constituent. Furthermore, this AI/ML constituent ODD is also used to estimate the generalization capabilities of the ML inference model and to define the behavior of the AI/ML constituent when exposed to out-of-distribution data. This AI/ML constituent ODD concept is now data-centric [6], and therefore, in its specification, data-specific considerations have to be included [27, 44,45]. These data-specific considerations can include how system-level parameters can be translated into parameters of the AI/ML constituent ODD [44] or how to represent the data distributions for the different parameters of the AI/ML constituent ODD [45]. In addition, different methodologies were introduced [27,44,45] to enable the identification and definition of parameters for an AI/ML constituent ODD that includes these data-specific considerations. Importantly, EUROCAE [46] and the G34 working group from SAE [47] are already developing standards using a similar notation of the AI/ML constituent ODD, as introduced by EASA. Most notably, ED-324 and ARP6983 [48], which will introduce a process for the certification and approval of AI in safety-related products. Both standards are expected to be first published in 2026 [49].

A possible application that is currently in focus for the development of AI-based systems in aviation is the airborne collision avoidance use case [35,37,38,50,51]. One system that is part of this use case is ACAS X. This system is designed to generate resolution advisories for pilots to follow to avert a potential mid-air collision between two aircraft. However, ACAS X relies on precalculated data for the resolution advisory generation, which, given its size, is a challenge to store on current and certified avionics hardware. Therefore, one approach is the utilization of neural networks to reduce the memory footprint of ACAS X [50,51]. However, the sole reliance on neural networks creates the issue that the memory footprint is not reduced loss-free [50,51]. Therefore, the Safety Net concept as a hybrid architecture for the compression of the data was suggested [34,51,52]. This concept's advantage is that

it can reduce the memory footprint while at the same time ensuring the correctness of the compressed data [34,51,52].

3. Methodology

For the development of the novel methodology, the considered objectives are grouped into five steps. These steps aim to fulfill the objectives of two blocks of the EASA trustworthiness guidelines, namely, the AI trustworthiness analysis and the AI assurance. The first three steps are the ConOps and OD definitions, as well as a functional decomposition of the system. Completing these steps should give the stakeholders of the system a clear understanding of the capabilities and limitations of the system, which is part of the AI trustworthiness analysis [5]. The next step is the definition of the AI/ML constituent ODD. This AI/ML constituent ODD must contain the ranges and distributions of the operating parameters for which the AI/ML constituent is designed to operate. The last step introduces guidance on how the specified AI/ML constituent ODD can influence the AI/ML constituent architecture, with a focus on the input feature selection for the constituent. While this step does not fulfill a certain objective, it supplies a necessary basis for other objectives that are part of the data preparation of the data management [5]. Importantly, these two steps are based on the artifacts produced by the steps of the AI trustworthiness analysis. Furthermore, these are part of the learning assurance that is part of the AI assurance framework, which is used for the development of the AI/ML constituent [5]. Following these steps should help the developers build an assurance case that the AI-based system complies with a defined level of performance and the defined requirements [5]. In summary, the main objective is the introduction of a methodology that bridges the gap between the AI trustworthiness analysis and the AI/ML constituent ODD that is part of the AI assurance. Importantly, using the proposed methodology, only a subset of all objectives of EASA's guidelines is fulfilled. Therefore, the proposed methodology will not be sufficient to build a complete assurance case for an AI-based system. Nevertheless, fulfilling the considered objectives is a mandatory prerequisite when building an assurance case according to EASA [5].

3.1. Definition of a Concept of Operations

The first step in the development of any AI-based system in aviation is the definition of said AI-based system through the description of the ConOps [5]. This step aims to identify all users of the system and to describe the capabilities and limitations of the system. Based on EASA's suggested objectives CO-01 and CO-02 for the system, all end users who interact with the AI-based system [5] must be identified and documented. Furthermore, this documentation also includes the goals and high-level tasks a user intends to perform when interacting with the AI-based system [5]. Following the identification of the users of the AI-based system, the next step is the definition of the ConOps for the AI-based system. The ConOps documents the characteristics of the AI-based system from the users' operational viewpoint [5]. Therefore, this methodology proposes the five steps for the definition of the ConOps based on the ISO/IEC/IEEE 29148:2018 [53]. These steps consist of a description of the current system, the justification for and nature of the changes, the description of the proposed AI-based system, the definition of the task allocation pattern, and lastly, the description of operational scenarios. The definition of the operational environment is replaced by the operational domain, which is defined in the following subsection.

As the first step, a description of the current system must be created [53]. This description must include an overview of the provided system functionality, an explanation of the underlying technology, and a list of aviation standards that apply to the system. This allows other stakeholders to better understand the current state of the problem domain [53], and also introduces the initial scope for the definition of the intended use of the system. As recommended in ISO/IEC/IEEE 29148:2018 [53], the justification of changes must highlight the shortcomings of the current system or situation. Furthermore, the nature of the proposed changes has to be stated [53]. Importantly, the change introduced with the AI component has to be stated to communicate the expected effect on the use of the AI components.

Based on these steps, a description of the AI-based system must be created. The description of the proposed AI-based system must contain “[t]he operational environment and its characteristics” [53]. Furthermore, it must also include the capabilities and functions provided by the proposed system [53]. This must also include the major system components needed for those capabilities and functions [53]. Lastly, the description also provides the task allocation and interactions between the end users and the AI-based system [5]. The description must enable all stakeholders to have a clear understanding of the functionality and responsibilities of the AI-based system. Moreover, the description must be extended by the description of operational scenarios. Each scenario description has to consist of multiple steps that describe the environmental conditions, the individual system functions, and the task allocation between the end user and the AI-based system that are needed to achieve the higher-level task or goal in the scenario [5]. In addition, these steps must be described in such a way that each stakeholder, regardless of technical background, can understand the system’s functionalities, responsibilities, and limitations. This is important, as this understanding of the AI-based system, especially from the end user’s perspective, will be the foundation for trust in the system [5]. Finally, these scenario descriptions must also include scenarios where the AI-based system is outside its designed operation conditions [5] to highlight the fallback measures and limitations of the AI-based system.

3.2. Definition of an Operational Domain

The OD of an AI-based system describes the operating conditions under which it is designed to function as expected [5]. Furthermore, the OD must be in accordance with the defined ConOps for the AI-based system [5]. EASA states that the capturing of the operational conditions for a system is already a practice in the aviation sector, but that this is not formalized enough for systems that will be AI-based [5]. While EASA formalized properties for the OD concept, it did not introduce guidelines on how to specify an OD in any formalized way. The automotive sector has already advanced the topic of capturing the operational conditions by standardizing different aspects [19,20,54] and has successfully used it for the development of a level 3 automated driving system [55]. Therefore, this approach utilizes ideas and concepts from other domains to build the OD concept for the aviation domain. To specify the OD for an AI-based system, Figure 2 introduces different concepts and their relations, building on the concepts and relationships of an OD in the automotive domain [56] but adapted so that the terminology fits the concepts and definitions of EASA. The concept of the taxonomy characterizes the operational conditions by classifying these through a set of attributes [20]. These attributes are used in the statements to define the individual ranges of the operational conditions, as shown at the bottom right of Figure 2. A statement describes an operational condition excluded or included in the OD specification. The collection of statements comprises the OD specification for the AI-based system, i.e., the operating conditions under which the AI-based system is expected to operate. Importantly, the OD is normally a subset of all possible operational conditions, as systems can be restricted to certain operational conditions due to their function and design [56]. For the statements that define the OD, a specification language is needed, as this allows the communication and understanding of the OD by different stakeholders [20]. Therefore, to specify an OD for an AI-based system, both an OD taxonomy and an OD specification language have to be defined [56]. The definition of an OD taxonomy is specific to the use case for which the AI-based system is built, while the OD specification language should be usable for any use case. These two components define a concrete OD for an AI-based system. The following sections will first introduce a hierarchical structure for the OD taxonomy and a definition language for the OD. Secondly, an approach to specifying a concrete OD for an AI-based system’s use case is outlined.

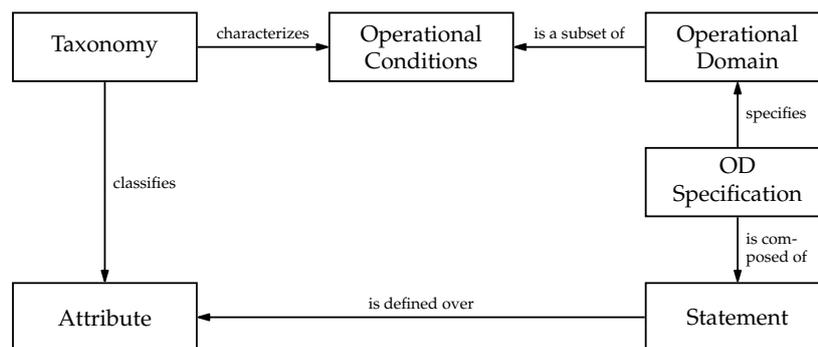


Figure 2. Relation between the different concepts connected to an operational domain. The figure is adapted from [56], which defined the relations and components for the automotive concept of an OD.

3.2.1. Operational Domain Taxonomy

An OD taxonomy defines the attributes that can make up the operating environment of the AI-based system and organizes them in a hierarchical structure [20]. Thus, the taxonomy must cover all possible attributes necessary to define the elements and conditions of the operating environment that affect the AI-based system. This is important, as an OD can only be considered complete if no safety-relevant attributes are missing [57]. Therefore, completeness is an important prerequisite to argue for a system's safety based on the OD [57]. However, due to the diversity of use cases for AI-based systems in the aviation sector [8,15,16], developing a single, universal taxonomy that accommodates all use cases remains a significant challenge. Therefore, while the individual domains may share similar taxonomies for different use cases, across domains, a single universal OD taxonomy is not feasible. Thus, the proposed methodology aims to define a generic approach to build a foundation for the OD taxonomy definition applicable to various use cases. For the basic structure of the OD taxonomy, three top-level attributes are proposed, namely *scenery elements*, *environmental elements*, and *dynamic elements* [20,58]. This proposed top-level attribute structure was already applied successfully for the use cases of air traffic management [32] and air operations [37]. The first top-level attribute, the scenery elements, should include all attributes that define the elements that can be considered spatially fixed in the operating environment of the AI-based system [20]. While these elements are fixed, it does not mean that they have to be in a static state. Next, the environmental elements should include all attributes that describe the weather, atmospheric conditions, and all other attributes that are considered to be non-scenery elements [20]. A non-scenery element can be, for example, the connectivity of the system with an external infrastructure [20]. Finally, the dynamic elements describe all attributes that describe other participants in the environment and the AI-based system itself [20]. The other participants can be seen as agents that can move or change in the operating environment of the AI-based system. Additionally, the dynamic elements should include attributes that may restrict the performance or capabilities of the AI-based system itself [20]. This outlined taxonomy classification only gives a high-level guideline on the different attributes the developer has to define for their use case. This high-level view is chosen as these guidelines must apply to a vast variety of use cases in the aviation domain, such as air traffic management or maintenance. Therefore, depending on the use case for some top-level attributes, no attributes might be identified. For the definition of an OD taxonomy, a variety of formats can be used, including, among others, textual descriptions [19,20] or block definition diagrams written in SysML [32,37].

3.2.2. Operational Domain Definition Language

The OD definition language is an important part of an OD definition as it enables developers to capture the operational conditions consistently, accurately, and understandably by different stakeholders. The latter is required, as EASA views the OD as a part of the ConOps, which has to be comprehensible from a user's perspective [5]. In the automotive domain, the OD definition language is already standardized by the ISO 34503:2023 [20]. Therefore, for the definition language, a syntax based on the tabular approach of the ISO 34503:2023 is introduced. This format was introduced as it allows

the OD to be readable by all stakeholders connected to the system. The tabular format consists of the columns of the top-level attribute, multiple possible levels of sub-attributes, the qualifier, and the attribute with its corresponding value and unit. The columns of the top-level attribute, the sub-attributes, and the attribute are based on the hierarchical taxonomy. The qualifier column describes whether an attribute and its values are excluded or included in the OD. If an attribute and its value are included, the system can function under the stated condition. This also includes the combination of all possible variations of other included attributes and their attribute values. If the qualifier is excluded, then the attribute and its value are a condition in which the system is specified not to function. Therefore, in these situations, the system is not allowed to be operated or must be deactivated. Attributes not listed in the specification of the OD are assumed not to affect the AI-based system [20]. These assumptions must be later validated to ensure their correctness. The value of the attribute is described in the column "Attribute value". Depending on the type of attribute, a vast variety of different styles to describe the value may be necessary. The advantage of this approach is that the tabular format is not only readable by humans but can also be easily translated into a machine-readable format.

3.2.3. OD Specification

For the specification of an OD, the *domain-agnostic and risk-based OD definition approach* [58] is used but adapted to fit the framework proposed by EASA [5]. The approach was chosen as it defines the OD definition process domain independently. This is an advantage compared to other processes [22,59], as these mainly focus on the automotive domain. As EASA specified certain aspects regarding the OD, these are incorporated into the existing process. Furthermore, the existing process is extended with additional guidance on the identification and selection of attributes and their value ranges. The introduced process consists of three steps: the initialization, the refinement, and the validation and verification. The starting point in the initialization is the definition of the taxonomy for the use case, following the outlined structure from [subsection 3.2.1](#). This initialization of the taxonomy is based on the ConOps for the AI-based system, as it provides a generic description of the system and gives concrete descriptions of scenarios the system will be operated in. Both of these descriptions can be used to identify the attributes of the taxonomy. For the identification of the different elements in the operational conditions for the individual scenarios, approaches similar to the 6-layer model [31], developed for the automotive domain, can be utilized. In the approach of the 6-layer model, the idea is to split the operating environment into the spatial layers of the road network, roadside structures, temporary modifications, and the temporal layers of the dynamic objects, environmental conditions, and digital information [31]. The idea is that the decomposition of the operating environment into the different layers allows a structural approach for the system developer to identify all relevant elements in the operating environment. What each layer describes has to be adapted for the individual use cases to the applied model. In addition, depending on the use case, standardization efforts may already provide an initial set of attributes for an OD taxonomy. When an initial taxonomy is defined, it can be used to define an OD specification for the system. If an initial OD taxonomy is defined, the next step is to assign values to all relevant attributes. For this initial assignment of values, the ConOps for the AI-based system should be used. An important part of the determination of an attribute range is the identification of the qualifier. This allows for determining whether an attribute from the taxonomy is relevant for the operation of the AI-based system and whether it is a range that is included or excluded from the operating conditions of the AI-based system.

If an initial OD is defined, it builds the basis for the refinement. This refinement consists of multiple possible tasks, similar to prior works [32,58]. Firstly, based on the individual attributes identified and established by scenarios from the ConOps, it should be determined if these attributes require clarification in the scenarios. This is required due to the close connection between the scenarios of the ConOps and the OD, and the requirement that the ConOps and OD must be consistent [5]. Secondly, standards that are connected to the use case are analyzed. Depending on the proposed AI-based system, standards may already have defined operational services and environment descriptions,

or minimum design and performance requirements. Thirdly, subject-matter experts can be used to refine the defined OD taxonomy or the OD specification.

Importantly, as the last step of this refinement, again, the consistency between the OD and ConOps must be checked, as newly identified attributes in the refinement might introduce additional operational conditions for the system. These newly identified operational conditions might have to be reflected in the operational scenario descriptions of the ConOps. This can lead to an adaptation of the already defined scenarios or to the introduction of new scenario descriptions that include the newly identified operational conditions. The need for adaptation is a bidirectional relationship between the ConOps and OD specifications. Therefore, if the ConOps is adapted, this induces adaptations required in the OD taxonomy or specification. Adaptations such as these are necessary, as otherwise inconsistencies would persist between the OD and ConOps for the AI-based system.

If the refinement is complete, a verification and validation of the specified OD is necessary. This verification and validation should ensure that the OD describes all necessary operating conditions to meet the requirements of the system and that the OD and ConOps are consistent.

3.3. Functional Decomposition of the AI-Based System

To achieve a functional decomposition of the AI-based system, a two-step approach is proposed. The first step is a functional analysis of the system, and the second step is the definition of the preliminary system architecture. At the end of the functional decomposition, an allocation of the functions to the AI/ML constituent should be achieved. The goal of the functional analysis is to identify the different functions that are implemented by the AI-based system. As described by the ARP4754 [1], a function should capture the behavior of a system, regardless of the chosen implementation. The functions of a system can be captured by a functional tree [60]. The functional tree decomposes the system into its basic functions. Here, the analysis starts with the high-level function of the system. This high-level function is then subdivided into lower-level functions [60]. These lower-level functions are further decomposed until an atomic function is reached, one that cannot be split further. These basic functions are necessary, as they build the foundation of the system's functional requirements. For the creation of the individual functions, a set of different rules is recommended [60]. Most importantly, these atomic functions should be described as general as possible, using a verb and a noun, so as not to restrict the variety of solutions [60]. In the second step, a preliminary system architecture is introduced. This includes an overview of all components in the system and shows the information flow between these components [5]. One specialty required by EASA when creating an AI-based system architecture is the labeling of the different components, functions, and items, whether they are AI/ML-based or not [5]. A function is AI/ML-based if it is implemented by an item that contains an AI/ML constituent. This classification is important to specify, as it highlights which parts of the system the AI assurance needs to be applied to. In addition, processes such as ARP4754 [1] or ARP4761 [61] can identify additional functional, safety, and security requirements. Based on the functional decomposition of the previous step and the additional identified requirements, a necessary foundation for the creation of the AI/ML constituent requirements is created. However, as noted in AIR6988 [4], the ARP4754 [1] guideline has gaps when assessing AI/ML-based systems. Therefore, when dealing with such systems, ARP4754 [1] might not be sufficient, and additional processes such as ARP6983 [48] have to be applied.

3.4. Definition of the AI/ML Constituent Operational Design Domain

The previous sections introduced the steps necessary for the proposed methodology to fulfill the required objectives at the system level. For the AI assurance, EASA [5] requires further refinement and allocation of the system-level requirements to the AI/ML constituent. An important concept at the level of the AI/ML constituent is the definition of the AI/ML constituent ODD. Furthermore, the AI/ML constituent ODD must also define constraints and requirements for the ML inference model and the data used to build and implement this model [5]. Lastly, the AI/ML constituent ODD must also incorporate constraints and requirements on the data that the ML inference model will be exposed to during inference operations [5]. As the OD and the ODD are conceptually similar, the

proposed methodology is built upon the OD framework introduced in [subsection 3.2](#). Importantly, due to some key differences between the OD and ODD, several adaptations must be incorporated into the methodology for the ODD definition to adhere to the requirements of EASA [5].

3.4.1. AI/ML Constituent ODD Taxonomy

Similar to the OD taxonomy at the system level, the AI/ML constituent ODD taxonomy defines the attributes of the operating environment for the AI/ML constituent [5]. Therefore, a similar approach to define the attributes for the AI/ML constituent ODD was chosen, as was used for the attribute definition of the OD. This includes the structuring of the operating conditions into the top-level attributes of *scenery elements*, *environmental elements*, and *dynamic elements*. As the operating conditions for the AI/ML constituent are specified at the subsystem level [5], influences stemming from the overall system can affect the provided data. As such, these influences must be captured in the AI/ML constituent ODD to ensure that the data conforms to the specified conditions [5,62]. Accordingly, the top-level attributes are extended with the top-level group of *operating parameters* [5]. This group of attributes must contain all the additional operating parameters introduced by other system components that the AI/ML constituent interacts with.

3.4.2. AI/ML Constituent ODD Definition Language

The definition language of the OD is also applied to the definition language of the AI/ML constituent ODD. Therefore, the same tabular structure as in [subsection 3.2.2](#) is used, but extended by two additional columns. Firstly, for each attribute, the distribution of the data must be defined [5]. Whenever possible, the system component, e.g., a sensor, from which the data originates, should also be recorded. This distribution column describes the distribution of the data under which the AI/ML constituent is expected to operate [5]. The distribution of each parameter describes the assumed underlying distribution, from which the data were sampled independently [5]. These distributions are important to record, as the generalization capability of an ML model is approximated based upon the assumption that the out-of-sample data is sampled from the same distribution as the in-sample data [5]. This is crucial, as the generalization capability of an ML model is a probabilistic statement that is only valid if the assumption of the same underlying distribution is met [15]. Thus, the distribution information is crucial to record as it can later be used in the ODD monitoring. Therefore, it is necessary to ensure that the ML model is only exposed to data distributions it was trained on, thereby enabling the AI/ML constituent to provide its intended behavior [5].

In addition to EASA requirements, it is proposed to incorporate a column to record the sensor, allowing for the identification of the data source within the system. This information can be used in data management to ensure the correctness of data, as the knowledge about the sensor allows for the identification of potential errors that might be introduced with the chosen sensor setup during data collection. Also, the inclusion of the sensors in the ODD allows for an assessment of the sensor setup [22]. This assessment of the sensor setup and its characteristics allows for determining if the sensor setup at the system level is sufficient for the defined AI/ML constituent ODD. Otherwise, an iteration and modification of the sensor setup at the system level might be required [22]. Importantly, it must be noted that not every ODD parameter must, or even can, be directly measured by a sensor; parameters can also be indirectly recorded, e.g., by inferring from other measurements. Therefore, these parameters might only have to be collected in the data acquisition phase to ensure that the specified conditions were captured. During operation, depending on the ML model, these parameters might not be directly used as input for the ML model in the training and in the inference during operation, and are only indirectly captured, for example, the weather conditions in an image. However, these attributes still have to be included in the AI/ML constituent ODD and in the collected dataset to allow a successful training of the ML model in those different specified conditions. Lastly, based on the sensor information and the safety assessment at the system level, it is possible, through processes such as ARP4761A [11], to additionally identify potential invalid or erroneous input data regions. Such an assessment can be used to identify scenarios and data that can be used to test the ML model's

behavior under these degraded conditions. This information can be used for the definition of the ODD monitoring capability of the system [5] and for the development of mitigation strategies [6].

3.4.3. AI/ML Constituent ODD Specification

The previous two sections, [subsubsection 3.4.1](#) and [subsubsection 3.4.2](#), defined the components for the ODD specification. As with the OD, additionally, a process is needed for a methodical development of an ODD specification for any use case or AI-based system. The proposed process is based on the *domain-agnostic and risk-based ODD definition approach* [58], which consists of the steps of initialization, refinement, and verification and validation of the AI/ML constituent ODD. The first step in the process is to initialize the AI/ML constituent ODD. This initialization is based on the functional decomposition, the OD, and the requirements allocated to the AI/ML constituent. It defines an ODD based on the system-level information. When the initial definition of the ODD is completed, the next step is to refine the ODD to incorporate AI/ML constituent-specific information and concepts. This refinement includes specifics about the sensor setup and the AI/ML constituent architecture, which have to be considered for the definition of the ODD. These considerations are focused on the incorporation of specifics that are necessary for the data management and learning process management of the AI assurance process [5]. This is required, as the AI/ML constituent ODD must provide “a framework for the selection, collection, [and] preparation of the data” [5] that is used for the development of the ML inference model [5]. The last step in the process is the verification and validation of the AI/ML constituent ODD. This verification and validation are outlined in the anticipated MOC DA-07 [5], where the verification and validation consist of subject-matter experts reviewing the operating parameters regarding correctness and completeness.

The initialization of the AI/ML constituent ODD is based on the ConOps, OD, functional decomposition, and requirements allocated to the AI/ML constituent. Therefore, the different artifacts from the AI trustworthiness analysis have to be used for the initialization of the AI/ML constituent ODD [5]. For each OD attribute, it should be determined whether it is a relevant operating condition for the AI/ML constituent, based on the functions and requirements allocated to it. To guide this identification, each parameter should be classified into a group of either a physical range or a parameter that affects the distribution of the data, such as the behavior of dynamic elements. If an attribute is classified as a physical range, based on the allocated function and requirements to the AI/ML constituent, this range can then be adapted accordingly. The second option is to classify an attribute as one that influences the distribution of the data. These attributes are important to recognize, as they will have a large impact on how the individual samples are distributed in the dataset used for the development of the ML inference model. Furthermore, in the data collection, these attributes have to be monitored to ensure that the conditions that these attributes imply will be included in the dataset. Using this classification for each attribute, it can later be determined if it is relevant for the AI/ML constituent and the data management. It must be noted that not every attribute will be distinguishable in one of these two groups, and therefore, this classification will depend on the subjective judgment of the developers. At the end of the initialization of the AI/ML constituent ODD, an initial ODD is created. The created ODD is a subset of the OD, as the initialization only relies on the attributes and their ranges that are introduced in the OD. This will likely change with the refinement of the ODD in the next stage of the process. Furthermore, it is important to note that this refinement can result in an AI/ML constituent ODD that is a superset of the system’s OD. Such a superset relationship can be used to improve a model’s performance and stability by ensuring it is trained on a broader range of available data [5].

As the initialized AI/ML constituent ODD is purely based on the OD, and the OD is not sufficient for the data management and learning process management [5], the next step is the refinement of the initialized ODD into a form that is usable for the data management and learning process management. To achieve this, the projection of attributes, the determination of sensor characteristics, the identification of domain-specific concepts, and the analysis of existing data sources are introduced. These steps aim to introduce a level of detail that will be sufficient for the design and development of the ML inference model contained in the AI/ML constituent.

As discussed previously, the attributes of the initial ODD are based on the system-level OD. However, these attributes can be perceived differently by the AI/ML constituent than they are defined at the system level, mostly due to the sensors selected in the system architecture. Therefore, the attributes and ranges of the initial AI/ML constituent ODD might have to be projected into the dimensional space, which can be perceived by the AI/ML constituent. One necessary projection of attributes is the transformation of physical attributes. These transformations can be simple unit conversions, but also more complex operations, if, for example, semantically defined attributes of the OD have to be described using concrete physical sensor values. Another possible necessary transformation can be the transformation of geometric attributes into data characteristic properties, where these properties depend on the available sensor [44]. Also, it is possible that these transformations have to be applied in reverse to construct geometric properties based on image-level features. Applying these different transformations to the different attributes of the initial ODD should yield an ODD that fits the perception of the AI/ML constituent and is usable in data management.

An important aspect of data management is the collected data used for the development of the ML inference model. It must be ensured that the characteristics of the data match the data encountered by the ML inference model during its application [41,62]. One important aspect are the properties of the data that are caused by the characteristics of the sensors and their setups, which are used in the system. These characteristics can be the orientation, type, and position of the sensor or specific properties that are determined based on the sensor. In general, these properties can include the resolution or signal-to-noise ratio of the sensor [63]. Such characteristics are important to describe, as it was shown that these can negatively impact the performance of ML inference models [64]. The identification of these sensor characteristics can be based on the sensor specifications and how the sensor is installed in the overall system.

As previously explained, the OD describes the operating conditions of the AI-based system, while the AI/ML constituent ODD describes the operating conditions of only its corresponding AI/ML constituent. This discrepancy in definition implies additional or different conditions compared to the defined OD, as the AI/ML constituent can be influenced differently by those various conditions. The additional conditions will largely depend on the perception, e.g., sensor setup, of the AI/ML constituent. Furthermore, as the AI/ML constituent ODD must provide “a framework for the selection, collection, preparation of the data” [5], the ODD should also incorporate specific concepts that impact the ability of the ML inference model to learn its designated function and can affect the performance of the ML inference model in operation [44]. Due to the lack of robust methods to verify the outputs of ML inference models during operation, it is often difficult to detect erroneous outputs and identify their root causes. Therefore, the identification of such concepts is important in the design of the AI/ML constituent ODD. To identify such attributes for the AI/ML constituent ODD, other works have proposed the identification and classification of necessary, supportive, irrelevant, and false-positive concepts [44]. This identification and classification help to determine important aspects of the operating environment, which can affect the ML model to learn its designated function successfully. However, for these concepts, the identification in the domain of the AI/ML constituent remains a challenge. As ML models, especially deep learning architectures, are designed to extract patterns and features by themselves, it is unclear to the developer which of the known or unknown patterns and features in the data have the highest significance on the ability of the ML model to learn its designated function. Nevertheless, such an identification is important to ensure that the dataset that is collected accurately reflects the later operating conditions of the AI/ML constituent. For the identification of the attributes based on these different concepts, an approach similar to the 6-layer model can be used, which was introduced in [subsubsection 3.2.3](#). Therefore, based on the same idea of the 6-layer model [31] that was used at the system level, a model of the operating conditions for the AI/ML constituent can be built. Importantly, the defined model at the system level can be used as a basis, but has to be extended based on the perception of the AI/ML constituent. Furthermore, for the identification of all relevant attributes in the model that may affect the signal recorded by the sensors, the path of the signal should

be mapped in this model [27]. Through thorough mapping, the different objects that interact with the signal can be discovered and described as an AI/ML constituent ODD attribute. Additionally, to accurately describe and build a model for a use case, an ontology-based domain model can be built [65]. Such a domain model represents the natural language domain knowledge in a graphical form [65]. The resulting graph consists of elements that can be classified into entities, relations, attributes, and values that describe the corresponding concepts of the real world [65]. Based on the entities and attributes of the domain model, attributes in the AI/ML constituent ODD can be defined. The values that are in relation to these entities and attributes in the domain model can then be used to define the value range of the corresponding AI/ML constituent ODD.

Depending on the use case, available data sources can be used to identify relevant aspects for the AI/ML constituent ODD [62]. Such data sources can include standards that apply to the use case, available specifications for sensors, or available datasets from the same or a similar use case. Similar to the definition of the OD, standards can already provide a list or collection of operating conditions under which the AI/ML constituent might have to operate. In the aviation sector, standards exist that specify operational services and environment definitions, or minimum aviation system performance for systems that are used to provide certain functionalities. The information or requirements that are defined in these documents can be transferred into attributes for the AI/ML constituent ODD. In addition to these standards, sensor characteristics can be identified based on the specifications that are provided for the sensors that are used in the AI-based system. The final source of AI/ML constituent ODD attributes considered are available datasets of the same or similar use cases [62]. These discovered datasets can be explored to find additional attributes for the AI/ML constituent ODD. The exploration of a dataset consists of the identification of properties and their relation in the dataset [66]. Importantly, such an exploration must be used to identify new attributes and not just be used to generate an AI/ML constituent ODD, as the available dataset might not fully represent the operating environment of the AI/ML constituent.

3.4.4. AI/ML Constituent ODD Verification and Validation

The last step in the AI/ML constituent specification process is an independent AI/ML constituent ODD verification and validation. As previously outlined, an insufficient AI/ML constituent ODD can pose a safety risk for the overall system [57]. The verification and validation of the correctness and completeness of the ODD must be done by subject-matter experts [5]. In addition to the expert judgment, this verification should ensure that all relevant domain standards that apply to the system are considered [57]. Furthermore, additional datasets should be used in this verification and validation. Similar to the discovery of new attributes using data sources, these additional datasets can be used to ensure that the attributes and their values consider all relevant conditions [57].

3.5. Model Architecture and Input-Feature Selection

Following the definition of the AI/ML constituent ODD, the next step in the process is the definition of the AI/ML constituent architecture and input selection based on this defined AI/ML constituent ODD. The AI/ML constituent architecture includes all components, including preprocessing, the ML inference model, and the post-processing necessary to provide the function allocated to the AI/ML constituent [5]. As the AI/ML constituent ODD is not the only influencing factor for the architecture, a mapping of the different influence factors is depicted in Figure 3. These influencing factors were identified based on the different objectives of EASA's concept paper [5]. As shown in Figure 3, the *Inputs* for the AI/ML constituent that are used in the ML model are influenced by the AI/ML constituent ODD, the type of data that is collected, and the architecture. The AI/ML constituent ODD determines the different characteristics of the data to be captured based on the different conditions required, based on the different AI/ML constituent ODD attributes and their ranges. However, the AI/ML constituent ODD does not define the data type or format that is collected. Therefore, the data can be structured or unstructured, which will then largely determine what type of inputs are feasible for an ML model. While all these can provide the necessary information that might

be described in the AI/ML constituent ODD, due to the different types of data that are available for the same ODD, different types of input features are possible. One important step for the definition of the inputs can be the application of feature engineering methodologies to extract the most useful input features [67]. However, these feature engineering steps might not be necessary if an ML model architecture is selected that incorporates feature-extracting components. The architecture for the ML model is not directly dependent on the defined AI/ML constituent ODD, it is only influenced indirectly through the collected data types. In addition, the requirements [41] and allocated functions to the AI/ML constituent have a major influence on the architecture. The selection of a suitable ML model architecture depends on the performance that an individual ML model architecture has achieved in the different experiments [68]. Most importantly, the different components and the ML model have to adhere to the allocated requirements and derived performance metrics.

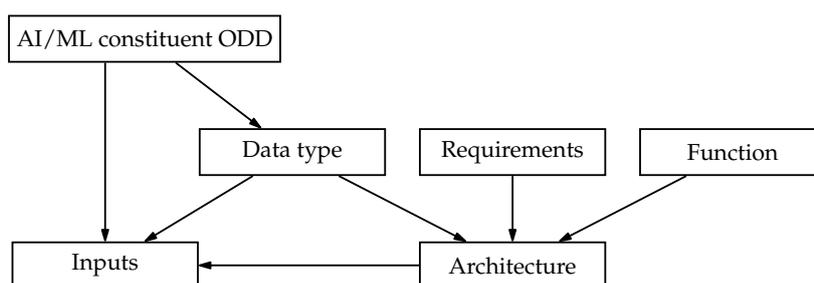


Figure 3. A mapping of the influence factors when designing an ML model.

4. Results

This section introduces the application of the proposed methodology for the selected airborne collision avoidance system use case. This application of the methodology aims at validating whether the application of the methodology conforms to the considered EASA objectives, see Table 1. For the use case, the vertical collision avoidance systems (VCAS) [69] and horizontal collision avoidance systems (HCAS) [70], both implemented in PYCASX [38,71], were selected. The PYCASX system is a Python-based implementation of HCAS and VCAS, inspired by ACAS Xa and ACAS Xu, but connected to the sophisticated FlightGear simulator [72–74]. This new system is proposed as a solution to reduce the memory footprint of ACAS Xa/Xu to make its implementation on avionics hardware feasible [38,51]. For the memory footprint reduction, neural networks are used due to their excellent compression capabilities [50]. But as the neural network compression is not loss-free, the Safety Net concept is proposed [34,51,52], which uses neural networks and sparse lookup tables to achieve a loss-free compression. As neural networks are utilized, PYCASX is an AI-based system. Therefore, the system has to conform to the considered objectives of the EASA concept paper [5]. For this reason, the methodology introduced in the previous section 3 is applied to PYCASX and its subsystems. Importantly, as PYCASX contains multiple subsystems that are not AI-based, the assumption is made that these subsystems will be built according to the relevant aviation guidelines and standards, such as DO-178C [2]. Therefore, in the following sections, these subsystems will only be introduced to the extent necessary for understanding the VCAS and HCAS subsystems and the overall functionality of the system.

4.1. Definition of a ConOps for PYCASX

The task of PYCASX is to provide partial ACAS capabilities to the pilots of an aircraft. For the definition of the ConOps, the following section exercises the individual steps defined previously in the methodology in subsection 3.1. Therefore, the first step is to provide a description of the current system. In the given use case, this can include the two systems, VCAS and HCAS, based on ACAS Xa and ACAS Xu, respectively. Each of these systems or components might potentially be replaced by PYCASX. ACAS Xa is standardized by the DO-385 [72] and ACAS Xu by the DO-386 [73]. Importantly, ACAS Xu, designed to provide resolution advisories for pilots in the vertical and horizontal plane,

is a superset of ACAS Xa. Thus, it is often sufficient to reason about ACAS Xu. In addition to these systems, further standards were identified that may apply to the ACAS use case, such as ED-271 [75] and ED-313 [76]. As previously introduced, the justification of the changes is based on the necessity to further reduce the memory footprint of ACAS Xu [51]. The description of the proposed AI-based system is based on the description provided by the initial implementation of PYCASX [38,71]. To summarize, PYCASX is designed to provide the pilots with last-resort measures to prevent mid-air collisions. The system is designed to operate in European airspace *type C*, where aircraft typically operate in the IFR or VFR mode. The airspace *type C* also entails that no geographic features or structures are present in the airspace. Both the ownship and all possible intruders must be equipped with an ADS-B system. Based on the information about the position and dynamics of the aircraft, the system can determine if the ownship is on a collision course with an intruder. If an intruder is on a collision course, the system shall generate a collision advisory for the pilot to prevent a potential collision. The collision advisories can either be on a vertical or horizontal plane. If multiple intruders are on a collision course, the system announces the strongest advisory to the pilot, as long as no conflicting advisories were calculated for the intruders. Furthermore, no coordination between the ownship and intruder is assumed. The pilots have to first decide whether the collision advisory can be executed safely and then execute the given collision advisory.

4.2. Definition of an OD for PYCASX

Following the introduction of the system capabilities and the operational description in subsection 4.1, the next step is to define the OD for the PYCASX system. As explained in the process of subsection 3.2.3, this consists of the OD taxonomy definition and the OD specification. For the identification of the elements of the OD, the recommended deconstruction of the environment using a layer model was applied [31]. The model consists of six layers, which are based on the automotive 6-layer model [31], and were adapted to fit the aviation use case to describe all elements that make up the airspace for en-route operations. The identified attributes and the resulting OD specification for the PYCASX system are shown in Table 2. For these defined attributes, values were already defined in the initialization based on the ConOps, and further refined using the identified standards that apply to PYCASX.

Table 2. The specified OD for PYCASX.

Top-level attribute	Sub-attribute	Qualifier	Attribute	Attribute value	Unit
Scenery	Airspace	Include	Type	C	-
	Airspace	Include	Flight Rule	IFR, VFR	-
	Airspace	Include	Altitude	[10 000, 66 000]	ft
	Airspace	Include	Latitude	[-90, 90]	°
	Airspace	Include	Longitude	[-180, 180]	°
	Airspace	Include	Route Type	Free Route Airspace	-
	Airspace	Exclude	Geography	Any	-
	Airspace	Exclude	Structures	Any	-
Environment	Weather	Exclude	Adverse Conditions	Any	-
	Connectivity	Include	Satellite Positioning	GPS	-
	Connectivity	Include	Communication Type	ADS-B	-
	Connectivity	Include	Communication Range	≥ 20	NM
Dynamic Elements	Intruder	Include	Agent Type	Airplane	-
	Intruder	Include	Maximum Agent Density	0.06	NM ⁻²
	Intruder	Include	Latitude	[-90, 90]	°
	Intruder	Include	Longitude	[-180, 180]	°
	Intruder	Include	Altitude	[10 000, 66 000]	ft
	Intruder	Include	Horizontal Airspeed	[0, 600]	kn
	Intruder	Include	Horizontal Acceleration	[-1.5, 1.5]	g
	Intruder	Include	Vertical Rate	[-5000, 5000]	ft min ⁻¹
	Intruder	Include	Vertical Rate Acceleration	$[-\frac{1}{3}, \frac{1}{3}]$	g
	Intruder	Include	Heading	[-180, 180]	°
Intruder	Include	Communication Type	ADS-B	-	

Table 2. Cont.

Top-level attribute	Sub-attribute	Qualifier	Attribute	Attribute value	Unit
	Ownship	Include	Agent Type	Airplane	-
	Ownship	Include	Latitude	[-90, 90]	°
	Ownship	Include	Longitude	[-180, 180]	°
	Ownship	Include	Altitude	[10 000, 66 000]	ft
	Ownship	Include	Horizontal Airspeed	[0, 600]	kn
	Ownship	Include	Horizontal Acceleration	[-1.5, 1.5]	g
	Ownship	Include	Vertical Rate	[-5000, 5000]	ft min ⁻¹
	Ownship	Include	Vertical Rate Acceleration	$[-\frac{1}{3}, \frac{1}{3}]$	g
	Ownship	Include	Heading	[-180, 180]	°
	Ownship	Include	Vertical Rate Capability	≥ 2000	ft min ⁻¹
	Ownship	Include	Vertical Rate Acceleration Capability	$\geq \frac{1}{3}$	g
	Ownship	Include	Turn Rate Capability	≥ 3	°s ⁻¹
	Ownship	Include	Turn Rate Acceleration Capability	≥ 1	°s ⁻²
	Ownship	Include	Pilot Type	Pilot, Remote Pilot	-

4.3. Functional Decomposition of PYCASX

Based on the introduced ConOps, a functional decomposition of the PYCASX was conducted. In the ConOps description, three main functions were identified. The first main function that was identified was the detection of other aircraft and their position. This function is split into three subfunctions necessary to sense the position, height, and velocity of the ownship and the surrounding intruders. The second main function determines if a collision with the intruder is imminent. This is further split into subfunctions to preprocess the data gathered by the first function, to calculate and determine the best possible advisory, and to resolve any conflicts in advisories if multiple are generated for numerous intruders. The third main function is the interface to the aircraft avionics. This main function consists of multiple subfunctions necessary to display and announce the chosen advisories to the pilots. Significantly, only the subfunctions in the second main function, which determine the best possible advisory, are AI/ML-based, while all other functions are traditional software and hardware items. Additionally, a preliminary system architecture is defined for the PYCASX system to implement these specified functions, as shown in Figure 4. The function of sensing the position of the intruder and ownship is implemented by the *ADS-B* and *GPS* in the presented architecture. The determination whether a collision is imminent is realized by the *CPA calculation*, *HCAS*, *VCAS*, and the *Multi-intruder advisory selection* components. Here *HCAS* generates advisories in the horizontal plane, while *VCAS* does so in the vertical plane. The *Display & Speaker* component is responsible for implementing the function to alert the pilots to the determined advisory.

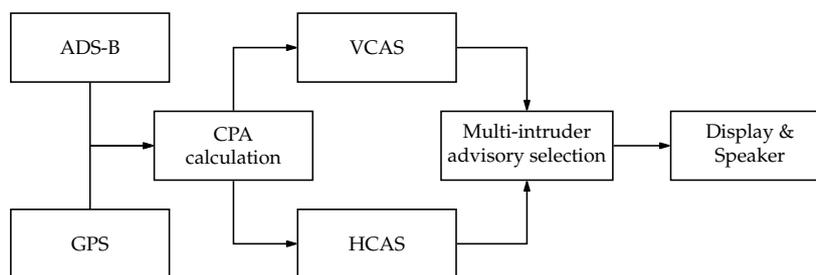


Figure 4. The preliminary architecture of PYCASX [38,71].

4.4. Definition of an AI/ML Constituent ODD for VCAS and HCAS

Following the previous steps for the PYCASX system, the next step for the development of the AI/ML constituents is to define an ODD for each AI/ML constituent, i.e., an AI/ML constituent ODD for VCAS and HCAS separately. This is necessary, as VCAS only provides advisories in the vertical plane and HCAS only in the horizontal plane. Therefore, these AI/ML constituents are created based on different datasets, and each dataset requires an individually defined AI/ML constituent ODD.

4.4.1. The AI/ML Constituent ODD for VCAS

Applying the methodology described in [subsubsection 3.4.3](#), the previously defined ConOps, OD, and functional decomposition of PYCASX are used in the definition of the AI/ML constituent ODD. As VCAS only provides vertical advisories, the AI/ML constituent ODD is only defined for the data in this plane.

Using the defined OD from [subsection 4.2](#), the first step is the identification of the individual parameters of the OD that must be part of the AI/ML constituent ODD. For this, as described in [subsubsection 3.4.3](#), all attributes were classified as attributes of a physical range or as behavior-determining attributes. Using this classification and the allocated function to VCAS for each attribute, it is determined if it is relevant for the AI/ML constituent ODD. For example, the OD attributes of the airspace, *altitude* and *coordinates* are both classified as physical ranges. As VCAS shall be able to resolve a conflict regardless of the airspace dimension, these are seen as not relevant for the VCAS ODD. Similar to the *type*, the *flight rule* and *route type* are both elements that determine the behavior of aircraft and are, therefore, relevant for the VCAS ODD. Other OD attributes, such as the *geography* and *structures*, are determined to be not relevant for the VCAS ODD as the detection of these conflicts is handled by other systems, for example, the ground proximity warning system [77]. The same steps were done for the attributes of *environment* and *dynamic elements*. Based on these results, an initial ODD for VCAS was defined, which was further refined in the following steps.

Following the definition of the initial ODD for VCAS, the next step is the projection of attributes into the perception of the VCAS system. In this case, this only affects the *altitude* attribute. With VCAS, it is not the absolute altitude of the ownship and the intruder that matters, but the difference in altitude. Therefore, for the data management, it will be important to collect data that contains sufficient samples for all possible relative altitudes between the ownship and the intruder. The maximum relative altitude (Δh_{\max}) between ownship and intruder is given by the maximum vertical rate (\dot{h}_{\max}) and the maximum time to the closest point of approach (τ) and therefore can be calculated based on the defined OD values.

The next step for the VCAS ODD is the identification of sensor characteristics. The only sensors used in PYCASX are GPS and ADS-B. ADS-B is the broadcast of information about the aircraft's position, speed, and altitude based on data from the global navigation satellite system (GNSS) [78]. For both the ownship and intruder, GNSS is assumed to utilize GPS; therefore, both are affected by the same sensor characteristics. As GPS directly provides the required information, the only additional sensor characteristic introduced is the inaccuracy of the GPS measurements. However, as GPS has a high accuracy [79] and at large heights, only uncommon causes, such as solar storms [80], cause larger inaccuracies; these inaccuracies are assumed to be zero.

To identify additional domain-specific concepts for VCAS, an ontology-based domain model was created for an encounter between an ownship and a single intruder. This ontology-based domain model was built according to the rules outlined in [subsubsection 3.4.3](#). As shown in [Figure 5](#), the model contains two entities, the *Intruder* and the *Ownship*. Both of these entities have the three attributes *Velocity*, *GNSS*, and *Position*. For the ownship one special attribute was identified, namely the *Pilot*, characterized by the attribute *Response Time*. This attribute describes the time the pilot needs from the announcement of the advisory to the start of the execution of the advisory. Only when the delay is accounted for, the system is able to generate advisories earlier to still allow enough time for a safe avoidance [81]. In this work, the reaction time is assumed to be 0 s, in line with previous research [38,50,51,82,83].

For VCAS, a dataset is available that was used to build the original version of VCAS [50]. Next to the advisories for the VCAS system, the dataset consists of only four different attributes for the encounter data. These attributes were already identified in previous steps; therefore, this dataset did not yield any additional attributes for the VCAS ODD. The final AI/ML constituent ODD for VCAS is shown in [Table 3](#). In total, 16 attributes were identified for the VCAS ODD. Three attributes are under the top-level attribute *Scenery*, eleven under *Dynamic Elements*, and two under *Operating Parameters*. For the *Environment* top-level attribute, no relevant attributes were discovered. Compared

to the PYCASX OD, the defined AI/ML constituent ODD for VCAS contains fewer attributes. These fewer attributes result from the allocated function and requirements to the AI/ML constituent, as the AI/ML constituent only has to cover a smaller subset of conditions compared to the conditions at the system level. For example, the VCAS component does not have to consider the attributes that define the operating conditions in the horizontal plane, as it is only designed to resolve conflicts in the vertical plane. The three attributes for the *Scenery* are all attributes that influence the behavior of the ownship and intruder. These are important to ensure that these different behaviors are included in the dataset. Under the *Dynamic Elements*, most of the identified attributes are physical ranges that determine the allowed distances and rates of the intruder and the ownship. A specialty for the ownship is the inclusion of attributes that require certain performance criteria for the ownship itself. These attributes are already included in OD but are also relevant for the AI/ML constituent ODD to ensure that the ownship conforms to the requirements needed to execute the advisories as expected. Where possible, for each attribute, it was recorded whether it is measured by a sensor in the system architecture. In the VCAS use case, this is either the *ADS-B* or *GPS* sensor. The *Distribution* was only determined for the attributes that are a constant, i.e., their range includes only a single value. For all other attributes, no distribution was determined, as their distribution will depend on the collected data. Therefore, this column can only be completed when the data is collected and used to build the final AI/ML constituent.

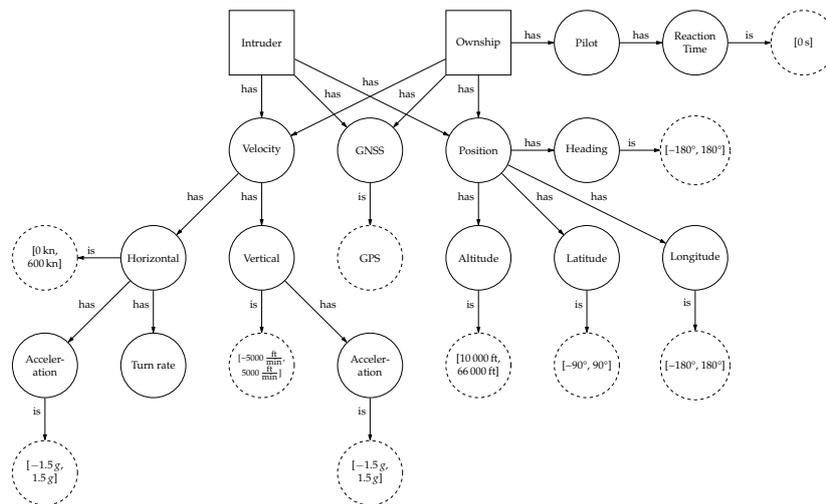


Figure 5. The complete ontology-based domain model for an encounter between the ownship and an intruder. Entities are depicted with a square, attributes by circles, values by dashed circles, and relations by arrows.

Table 3. AI/ML constituent ODD for VCAS.

Top-level attribute	Sub-attribute	Qualifier	Attribute	Attribute value	Unit	Distribution	Source
Scenery	Airspace	Include	Type	C	-	Constant	-
	Airspace	Include	Flight Rule	IFR, VFR	-	-	-
	Airspace	Include	Route Type	Free Route Airspace	-	Constant	-
Environment	-	-	-	-	-	-	-
Dynamic Elements	Intruder	Include	Agent Type	Airplane	-	Constant	ADS-B
	Intruder	Include	Vertical Rate	[-5000, 5000]	ft min ⁻¹	-	ADS-B
	Intruder	Include	Vertical Rate Acceleration	[- $\frac{1}{3}$, $\frac{1}{3}$]	g	-	-
	Intruder	Include	Relative Altitude to Ownship	[-10 000, 10 000]	ft	-	-
	Intruder	Include	Time until loss of separation	[0, 60]	s	-	-
	Ownship	Include	Agent Type	Airplane	-	Constant	-
	Ownship	Include	Vertical Rate	[-5000, 5000]	ft min ⁻¹	-	GPS
	Ownship	Include	Vertical Rate Acceleration	[- $\frac{1}{3}$, $\frac{1}{3}$]	g	-	GPS
	Ownship	Include	Vertical Rate Capability	≥2000	ft min ⁻¹	-	-
	Ownship	Include	Vertical Rate Acceleration Capability	≥ $\frac{1}{3}$	g	-	-
Operating Parameters	Ownship	Include	Pilot reaction time	[0]	s	Constant	-
	Intruder	Include	GPS Inaccuracy	None	-	Constant	GPS

4.4.2. The AI/ML Constituent ODD for HCAS

As HCAS is similar to VCAS, in the following section, the AI/ML constituent ODD for HCAS is introduced, but only in a shortened form that highlights the differences from the VCAS ODD definition. The initialization of the ODD for HCAS uses a similar approach compared to VCAS, with the only difference that for the *Dynamic Elements*, all attributes in the vertical plane are discarded and only the attributes in the horizontal plane are kept. As HCAS also relies on the same sensors as VCAS, the same additional attributes of the GPS accuracy were identified for the *Operating Parameters*. The identification of domain-specific concepts for HCAS was carried out using the same ontology-based domain model as for VCAS. Similar to VCAS, this model only provided the *Response Time* of the pilot as an additional attribute. The finalized AI/ML constituent ODD for HCAS is shown in Table 4. As already described, the major difference between the VCAS ODD and the defined HCAS ODD is that the encounters are on the horizontal plane. Therefore, the attributes of the *Scenery*, the *Environment*, and the *Operating Parameters* are the same as in the defined VCAS ODD. Only for the top-level attribute of *Dynamic Elements*, new attributes were introduced to describe the encounter geometry, and attributes of the vertical plane were discarded. This also includes attributes that ensure that the ownship executes these advisories in the dataset as specified. Therefore, the attributes of the *Turn Rate Capability* and the *Turn Rate Acceleration Capability* are included in the HCAS ODD.

Table 4. AI/ML constituent ODD for HCAS.

Top-level attribute	Sub-attribute	Qualifier	Attribute	Attribute value	Unit	Distribution	Source
Scenery	Airspace	Include	Type	C	-	Constant	-
	Airspace	Include	Flight Rule	IFR, VFR	-	-	-
	Airspace	Include	Route Type	Free Route Airspace	-	Constant	-
Environment	-	-	-	-	-	-	-
Dynamic Elements	Intruder	Include	Agent Type	Airplane	-	Constant	ADS-B
	Intruder	Include	Horizontal Airspeed	[0, 600]	kn	-	ADS-B
	Intruder	Include	Horizontal Acceleration	[-1.5, 1.5]	g	-	-
	Intruder	Include	Relative Angle to Ownship	[-180, 180]	°	-	-
	Intruder	Include	Time until loss of separation	[0, 60]	s	-	-
	Ownship	Include	Agent Type	Airplane	-	Constant	-
	Ownship	Include	Horizontal Airspeed	[0, 600]	kn	-	GPS
	Ownship	Include	Horizontal Acceleration	[-1.5, 1.5]	g	-	GPS
	Ownship	Include	Distance to Intruder	[0, 122 000]	ft	-	-
	Ownship	Include	Angle to Intruder	[-180, 180]	°	-	-
	Ownship	Include	Turn Rate Capability	≥ 3	$^{\circ} s^{-1}$	-	-
	Ownship	Include	Turn Rate Acceleration Capability	≥ 1	$^{\circ} s^{-2}$	-	-
	Ownship	Include	Pilot reaction time	[0]	s	Constant	-
	Operating Parameters	Ownship	Include	GPS Inaccuracy	None	-	Constant
Intruder		Include	GPS Inaccuracy	None	-	Constant	GPS

4.5. Framework Validation

The goal of the introduced methodology is to enable the developers of an AI-based system to fulfill the objectives introduced by EASA. For the methodology, a subset of all of EASA's objectives was selected. These selected objectives then imposed the conditions that have to be fulfilled by the individual steps in the methodology. They were later verified in Table 5 by the application to the PYCASX use case. The methodology was able to fulfill 72.2 % of the considered objectives requirements completely. No requirements were unfulfilled, and 27.7 % of the considered objective requirements were only partially fulfilled. Those were not fulfilled completely, as either the methodology only achieved results in the use case that covered the requirements indirectly, or the results depended on later steps of the W-shaped learning assurance process and, therefore, can only be fulfilled at the end of the development. Nevertheless, as most of the requirements are fulfilled, the verification of the methodology, using the PYCASX use case, is seen as successful. Furthermore, as shown in the previous sections, the methodology extended and concretized different concepts of the EASA concept paper, such as the OD and AI/ML constituent ODD. Importantly, as shown in the steps of defining the ODD for VCAS and HCAS, an approach was demonstrated to get from the system level to the AI/ML constituent level. Thus, the application of the methodology to the PYCASX use case confirmed

the benefits of the introduced formalisms and processes required by a developer, thus completing the necessary validation of the methodology.

Table 5. Identified and considered requirements of EASA's concept paper [5], which the methodology application must fulfill.

Objective	Requirement	Fulfilled
CO-01	Identification of end users	Partially
CO-02	Goals of the end users	Partially
	High-level tasks of the end users	Completely
CO-04	Operational scenarios	Completely
	Task allocation in the operational scenarios	Completely
	Capturing of operating conditions	Completely
CO-06	Functional decomposition of the system	Completely
	Function allocation in the system architecture	Completely
	Classification of AI/ML items	Completely
DA-03	Set of parameters for the AI/ML constituent	Partially
	Traced parameters to the OD	Completely

5. Discussion

In the previous chapter, the feasibility of the proposed methodology to fulfill the different requirements based on the considered objectives, see Table 1, was shown. With this novel methodology, it was possible to fulfill most of the objectives for the selected use case of PYCASX and the corresponding AI/ML constituents VCAS and HCAS. The failure to fulfill all requirements of the objectives was since the proposed methodology only covered a subset of all steps in the development of an AI-based system. For example, for the AI/ML constituent ODD, not for every attribute, a distribution was determined, as these distributions depend on the collected datasets. These datasets can only be determined after the AI/ML constituent is fully developed according to EASA's guidelines. Nevertheless, despite not completely meeting all requirements in the methodology's application to the use case, the methodology still showed its advantage of consolidating different approaches and concepts into one unified AI Engineering approach. In the methodology for the specification of the OD and AI/ML constituent ODD, a tabular format was chosen. While this has the advantage of being easily readable by different stakeholders, but lacks an abstract syntax that formalizes the metamodel of the OD and ODD. This could be improved by utilizing a modeling language, presumably SysML v2, to define an abstract syntax and a concrete syntax for the OD and AI/ML constituent ODD.

The advantage of the introduced methodology can be exemplified when comparing the results of the defined AI/ML constituent ODD with other AI/ML constituent ODDs that were specified for the same use case. One example of an ODD, defined by the MLEAP consortium [17] for the same use case of the airborne collision avoidance system, is shown in Figure 6. The specified ACAS Xu ODD only contains six attributes and their ranges. The HCAS ODD of this work is the equivalent to the shown ACAS Xu ODD. Compared to the ODD specified in this work, the MLEAP ACAS Xu ODD lacks the attribute classifications, the units, the qualifier, the attribute sources, and the distributions for the attributes. The latter is significant, as this is a requirement for an AI/ML constituent ODD according to EASA [5]. Therefore, the specified ACAS Xu ODD, as shown in Figure 6, does not fulfill the requirements of EASA. Furthermore, while the specified HCAS ODD contains 16 attributes, the MLEAP ACAS Xu ODD only contains six attributes. The six attributes are all contained in the HCAS ODD, and the major difference in attributes are the attributes that determine the behavior of the ownship and intruder, such as the *Turn Rate Capability*, see Table 4. However, it is necessary to include these attributes in the AI/ML constituent ODD, as they determine different behavior characteristics that can change the optimal advisories in different scenarios. Therefore, excluding these attributes will lead to an insufficient AI/ML constituent ODD that does not accurately describe the operating conditions of the AI/ML constituent. A sufficient level of detail is important, as only if the AI/ML

constituent ODD accurately describes the conditions the AI/ML constituent can encounter, a safety argument can be built upon the AI/ML constituent ODD for the system, allowing for later certification of the system [57]. The missing parameters in the MLEAP ACAS Xu ODD are also a problem for the AI/ML constituent ODD monitoring, as they are not included in the AI/ML constituent ODD and therefore will not be monitored. This becomes an issue because it cannot be adequately determined if the AI/ML constituent was designed to operate in these scenarios. Furthermore, as the AI/ML constituent ODD is also the framework for the selection, collection, and preparation of the data used to develop the ML inference model, the completeness of the data also depends on the defined AI/ML constituent ODD [5]. For example, the parameter of the *Agent Type* is included in the HCAS ODD but is missing in the ACAS Xu ODD. This allows for all types of agents, such as rotorcraft, to be included in the dataset, while the HCAS ODD is limited to airplanes. This again shows the necessity for an adequate AI/ML constituent ODD specification process. As introduced in section 2, the guidance for level 1 and 2 machine learning applications by EASA was only released in 2024, and the guidance for level 3 is to be expected in the upcoming years. Therefore, ambiguities and gaps remain in the currently available guidance [84]. For example, other works also identified issues or areas of improvement in certain AI assurance objectives [84]. The primary challenge encountered in the development of the methodology of this work was the ambiguity that existed in some of the objectives under consideration. One example is the definition of the OD by EASA. The OD is defined similarly to the AI/ML constituent ODD, in that it should describe the operating conditions at the system level [5]. However, in their definition, it is not clear if the idea of an OD is solely based on already existing practices in the aviation sector or if the OD is also based on practices of other industries, such as the automotive sector. In this work, the latter interpretation was chosen. Also, for the AI/ML constituent ODD, it is unclear if EASA permits the specification of the AI/ML constituent ODD based on collected data or if the AI/ML constituent ODD only relies on a top-down allocation approach of requirements and functions. Again, the latter interpretation was chosen.

$$ODD_{ACAS\ Xu} = \begin{cases} \tau \in [0, 101] \\ \theta \in [-3.14, 3.14] \\ v_{int} \in [0, 1200] \\ v_{own} \in [50, 1200] \\ \psi \in [-3.14, 3.14] \\ \rho \in [499, 185\ 318] \end{cases}$$

Figure 6. The ACAS Xu ODD defined in the MLEAP report [17]. From top to bottom, the individual attributes describe the time to loss of separation in seconds, the relative angle to the intruder, the speed of the intruder in feet per minute, the speed of the ownship in feet per minute, the intruder heading relative to the ownship, and the distance of the intruder to the ownship in feet.

6. Conclusions

This work addressed the issue of specifying the operating conditions of an AI-based system to comply with potential future EASA regulations. Therefore, a methodology was introduced that is built according to the objectives outlined by EASA's current guidelines. The methodology includes formats to define the OD, functional decomposition, and the AI/ML constituent ODD. Especially for the AI/ML constituent ODD, additions to this format were made to incorporate the data-specific conditions, as required by EASA. In addition to the formats required for the concepts, processes were introduced for the application of these concepts to concrete use cases. Importantly, a process was introduced that derives the AI/ML constituent ODD based on the OD and the description of the system. Furthermore, a mapping of the individual factors that influence the AI/ML constituent and ML model input and architecture was created. The methodology was applied to an airborne collision avoidance use case to validate its capabilities, where the goal is to introduce a neural network-based compression. Based on the use case, a ConOps, an OD, a functional decomposition, and an AI/ML

constituent ODD were specified for the PYCASX system according to the defined methodology. It was verified that most of the requirements of EASA's objective were satisfied for the chosen use case.

As discussed in [section 5](#), this work provided a baseline for the definition of the ConOps, OD, and AI/ML constituent ODD, but there are also gaps and areas of possible improvement. Firstly, the currently provided processes and guidance are mostly conceptual and need to be further refined and concretized. This concretization can take the form of developing a tool and an abstract syntax with which the OD and AI/ML constituent ODD can be defined. Secondly, the current processes and the conducted validation only considered the specification of the operating conditions for a system; therefore, the next step required is the definition of processes for the data management based on the defined AI/ML constituent ODD. Lastly, as the introduced processes currently only use the available EASA guidance for level 1 and 2 machine learning applications, adaptations of the processes might be necessary based on the upcoming guidance for level 3. Furthermore, the proposed processes should be applied to more use cases to validate their usefulness outside of the chosen use case. In particular, a more data-driven use case, especially one not yet governed by existing standards, could further test and refine the methodology. Moreover, a use case of level 3 could be chosen to find gaps in the current processes. Such an application could also serve to provide feedback to EASA in the development of their level 3 machine learning guidance.

Author Contributions: Conceptualization, F.W, J.C., and T.S.; methodology, F.W, J.C., and T.S.; validation, F.W.; investigation, F.W.; resources, J.C.; visualization, F.W.; writing—original draft preparation, F.W.; writing—review and editing, J.C., T.S., E.H., and S.H.; supervision, S.H., and F.K.; project administration, J.C.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: For this work, no new data were created.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ACAS	Airborne Collision Avoidance System
ADS-B	Automatic Dependent Surveillance–Broadcast
AI	Artificial Intelligence
AIR	Aerospace Information Report
ARP	Aerospace Recommended Practice
ConOps	Concept of Operations
CPA	Closest Point of Approach
DO	Document
EASA	European Union Aviation Safety Agency
ED	EUROCAE Document
EUROCAE	European Organization for Civil Aviation Equipment
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HCAS	Horizontal Collision Avoidance System
IFR	Instrument Flight Rules
ML	Machine Learning
NM	Nautical Mile
OD	Operational Domain
ODD	Operational Design Domain
SAE	SAE International
SysML	Systems Modeling Language
VCAS	Vertical Collision Avoidance System
VFR	Visual Flight Rules

References

1. S-18 Aircraft and Sys Dev and Safety Assessment Committee. *Guidelines for Development of Civil Aircraft and Systems*, 2010. <https://doi.org/10.4271/ARP4754A>.
2. RTCA, Inc.. Software Considerations in Airborne Systems and Equipment Certification. techreport DO-178C, GlobalSpec, Washington, DC, USA, 2012.
3. RTCA, Inc.. DO-200B - Standards for Processing Aeronautical Data. Technical report, GlobalSpec, Washington, DC, USA, 2015.
4. G-34 Artificial Intelligence in Aviation. *Artificial Intelligence in Aeronautical Systems: Statement of Concerns*, 2021. <https://doi.org/10.4271/AIR6988>.
5. European Union Aviation Safety Agency (EASA). EASA Concept Paper: Guidance for Level 1 & 2 Machine Learning Applications. techreport, European Union Aviation Safety Agency (EASA), Postfach 10 12 53, 50452 Cologne, Germany, 2024.
6. Kaakai, F.; Adibhatla, S.; Pai, G.; Escorihuela, E. Data-centric Operational Design Domain Characterization for Machine Learning-based Aeronautical Products. In Proceedings of the International Conference on Computer Safety, Reliability, and Security. Springer, 09 2023, pp. 227–242.
7. Mamalet, F.; Jenn, E.; Flandin, G.; Delseny, H.; Gabreau, C.; Gauffriau, A.; Beaudouin, B.; Ponsolle, L.; Alecu, L.; Bonnin, H.; et al. White Paper Machine Learning in Certified Systems. Research report, IRT Saint Exupéry ; ANITI, 2021.
8. European Union Aviation Safety Agency (EASA). Artificial Intelligence Roadmap 2.0. techreport, European Union Aviation Safety Agency (EASA), Postfach 10 12 53, 50452 Cologne, Germany, 2023.
9. European Union Aviation Safety Agency. The Agency. Available online: <https://www.easa.europa.eu/en/the-agency/the-agency> (accessed on 2024-08-15).
10. Durand, J.G.; Dubois, A.; Moss, R.J. Formal and Practical Elements for the Certification of Machine Learning Systems. In Proceedings of the 2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC), 2023, pp. 1–10. <https://doi.org/10.1109/DASC58513.2023.10311201>.
11. S-18 Aircraft and Sys Dev and Safety Assessment Committee. *Guidelines for Conducting the Safety Assessment Process on Civil Aircraft, Systems, and Equipment*, 2023. <https://doi.org/10.4271/ARP4761A>.
12. RTCA, Inc.. Design Assurance Guidance for Airborne Electronic Hardware. techreport, GlobalSpec, Washington, DC, USA, 2000.
13. Luettig, B.; Akhiat, Y.; Daw, Z. ML meets aerospace: challenges of certifying airborne AI. *Frontiers in Aerospace Engineering* 2024, 3. <https://doi.org/10.3389/fpace.2024.1475139>.
14. Federal Aviation Administration. Roadmap for Artificial Intelligence Safety Assurance. Technical report, Federal Aviation Administration, 800 Independence Avenue, SW, Washington, DC 20591, 2024.
15. EASA and Daedalean. Concepts of Design Assurance for Neural Networks (CoDANN). resreport, European Union Aviation Safety Agency (EASA), 2020.
16. EASA and Daedalean. Concepts of Design Assurance for Neural Networks (CoDANN) II with appendix B. resreport, European Union Aviation Safety Agency (EASA), 2024.
17. MLEAP Consortium. EASA Research – Machine Learning Application Approval (MLEAP) Final Report. Technical Report 1, European Union Aviation Safety Agency (EASA), 2024.
18. Bello, H.; Geißler, D.; Ray, L.; Müller-Divéky, S.; Müller, P.; Kittrell, S.; Liu, M.; Zhou, B.; Lukowicz, P. Towards certifiable AI in aviation: landscape, challenges, and opportunities, 2024.
19. The British Standards Institution. PAS 1883:2020 - Operational Design Domain (ODD) taxonomy for an automated driving system (ADS) – Specification; BSI Standards Limited, 2020.
20. ISO. ISO 34503: Road Vehicles – Test scenarios for automated driving systems – Specification for operational design domain; Beuth Verlag: Berlin, Germany, 2023.
21. On-Road Automated Driving (ORAD) Committee. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, 2021. https://doi.org/10.4271/J3016_202104.
22. Eichenseer, F.; Sarkar, S.; Shakeri, A. A Systematic Methodology for Specifying the Operational Design Domain of Automated Vehicles. In Proceedings of the 2024 IEEE 35th International Symposium on Software Reliability Engineering Workshops (ISSREW), Tsukuba, Japan, 10 2024; pp. 13–18. <https://doi.org/10.1109/ISSREW63542.2024.00040>.
23. Thorn, E.; Kimmel, S.C.; Chaka, M.; Hamilton, B.A.; et al. A Framework for Automated Driving System Testable Cases and Scenarios. Technical report, United States. Department of Transportation. National Highway Traffic Safety Administration, Washington, DC, USA, 2018.

24. Hallerbach, S. Simulation-Based Testing of Cooperative and Automated Vehicles. phdthesis, Carl von Ossietzky Universität Oldenburg, 2020.
25. PEGASUS RESEARCH PROJECT. The PEGASUS Method, 2019.
26. Fjortoft, K.; Rødseth, Ø. Using the operational envelope to make autonomous ships safer. In Proceedings of the The 30th European Safety and Reliability Conference, Venice, Italy, 2020.
27. Picard, S.; Chapdelaine, C.; Cappi, C.; Gardes, L.; Jenn, E.; Lefevre, B.; Soumarmon, T. Ensuring Dataset Quality for Machine Learning Certification. In Proceedings of the 2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), 2020, pp. 275–282. <https://doi.org/10.1109/ISSREW51248.2020.00085>.
28. Czarnecki, K. Operational world model ontology for automated driving systems–part 1: Road structure. Technical report, Waterloo Intelligent Systems Engineering (WISE) Lab, University of Waterloo, 2018. <https://doi.org/http://dx.doi.org/10.13140/RG.2.2.15521.30568>.
29. Czarnecki, K. Operational world model ontology for automated driving systems–part 2: Road users, animals, other obstacles, and environmental conditions,. Technical report, Waterloo Intelligent Systems Engineering (WISE) Lab, University of Waterloo, 2018. <https://doi.org/http://dx.doi.org/10.13140/RG.2.2.11327.00165>.
30. Mendiboune, L.; Benzagouta, M.L.; Gruyer, D.; Sylla, T.; Adedjouma, M.; Hedhli, A. Operational Design Domain for Automated Driving Systems: Taxonomy Definition and Application. In Proceedings of the 2023 IEEE Intelligent Vehicles Symposium (IV), 2023, pp. 1–6. <https://doi.org/10.1109/IV55152.2023.10186765>.
31. Scholtes, M.; Westhofen, L.; Turner, L.R.; Lotto, K.; Schuldes, M.; Weber, H.; Wagener, N.; Neurohr, C.; Bollmann, M.; Körtke, F.; et al. 6-Layer Model for a Structured Description and Categorization of Urban Traffic and Environment. *IEEE Access* **2021**, *9*, 59131–59147. <https://doi.org/10.1109/ACCESS.2021.3072739>.
32. Stefani, T.; Jameel, M.; Gerdes, I.; Hunger, R.; Bruder, C.; Hoemann, E.; Christensen, J.M.; Girija, A.A.; Köster, F.; Krüger, T.; et al. Towards an Operational Design Domain for Safe Human-AI Teaming in the Field of AI-Based Air Traffic Controller Operations. In Proceedings of the 2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC), San Diego, CA, USA, 09 2024; pp. 1–10. <https://doi.org/10.1109/dasc62030.2024.10749684>.
33. Berro, C.; Deligiannaki, F.; Stefani, T.; Christensen, J.M.; Gerdes, I.; Köster, F.; Hallerbach, S.; Raulf, A. Leveraging Large Language Models as an Interface to Conflict Resolution for Human-AI Alignment in Air Traffic Control. In Proceedings of the 2025 AIAA DATC/IEEE 44th Digital Avionics Systems Conference (DASC), Montreal, Canada, 09 2025; pp. 1–10.
34. Gabreau, C.; Gauffriau, A.; Grancey, F.D.; Ginestet, J.B.; Pagetti, C. Toward the certification of safety-related systems using ML techniques: the ACAS-Xu experience. In Proceedings of the 11th European Congress on Embedded Real Time Software and Systems (ERTS 2022), Toulouse, France, 06 2022; pp. 1–11.
35. Stefani, T.; Anilkumar Girija, A.; Mut, R.; Hallerbach, S.; Krüger, T. From the Concept of Operations towards an Operational Design Domain for safe AI in Aviation. In Proceedings of the DLRK 2023, Stuttgart, Germany, 09 2023; pp. 1–8.
36. Stefani, T.; Christensen, J.M.; Hoemann, E.; Anilkumar Girija, A.; Köster, F.; Krüger, T.; Hallerbach, S. Applying Model-Based System Engineering and DevOps on the Implementation of an AI-based Collision Avoidance System. In Proceedings of the 34th Congress of the International Council of the Aeronautical Sciences (ICAS), Florence, Italy, 09 2024; pp. 1–12.
37. Anilkumar Girija, A.; Christensen, J.M.; Stefani, T.; Hoemann, E.; Durak, U.; Köster, F.; Hallerbach, S.; Krüger, T. Towards the Monitoring of Operational Design Domains Using Temporal Scene Analysis in the Realm of Artificial Intelligence in Aviation. In Proceedings of the 2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC), San Diego, CA, USA, 09 2024; pp. 1–8. <https://doi.org/10.1109/dasc62030.2024.10749445>.
38. Christensen, J.M.; Anilkumar Girija, A.; Stefani, T.; Durak, U.; Hoemann, E.; Köster, F.; Krüger, T.; Hallerbach, S. Advancing the AI-Based Realization of ACAS X Towards Real-World Application. In Proceedings of the 2024 IEEE 36th International Conference on Tools with Artificial Intelligence (ICTAI), Herdon, VA, USA, 10 2024; pp. 57–64. <https://doi.org/10.1109/ICTAI62512.2024.00017>.
39. Torens, C.; Juenger, F.; Schirmer, S.; Schopferer, S.; Zhukov, D.; Dauer, J.C. Ensuring Safety of Machine Learning Components Using Operational Design Domain. In Proceedings of the AIAA SciTech 2023 Forum, 2023, p. 1124.
40. Gariel, M.; Shimanuki, B.; Timpe, R.; Wilson, E. Framework for Certification of AI-Based Systems, 2023.

41. Hasterok, C.; Stompe, J.; Pfrommer, J.; Usländer, T.; Ziehn, J.; Reiter, S.; Weber, M.; Riedel, T. PAISE®. Das Vorgehensmodell für KI-Engineering. Technical report, Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung (IOSB), 2021.
42. Zhang, R.; Albrecht, A.; Kausch, J.; Putzer, J. H.; Geipel, T.; Halady, P. DDE process: A requirements engineering approach for machine learning in automated driving. In Proceedings of the 2021 IEEE 29th International Requirements Engineering Conference (RE). IEEE, 2021, pp. 269–279.
43. Christensen, J.M.; Stefani, T.; Anilkumar Girija, A.; Hoemann, E.; Vogt, A.; Werbil, V.; Durak, U.; Köster, F.; Krüger, T.; Hallerbach, S. Formulating an Engineering Framework for Future AI Certification in Aviation. *Aerospace* 2025, 12, 1–27. <https://doi.org/10.3390/aerospace12060482>.
44. Cappelletti, C.; Cohen, N.; Ducoffe, M.; Gabreau, C.; Gardes, L.; Gauffriau, A.; Ginestet, J.B.; Mamalet, F.; Mussot, V.; Pagetti, C.; et al. How to design a dataset compliant with an ML-based system ODD? In Proceedings of the 12th European Congress on Embedded Real Time Software and Systems (ERTS), Toulouse, France, 06 2024; pp. 1–10. <https://doi.org/10.48550/ARXIV.2406.14027>.
45. Höhdorf, L.; Dmitriev, K.; Vasudevan, J.K.; Subedi, S.; Klarmann, N.; Holzapfel, F. Artificial Intelligence Verification Based on Operational Design Domain (ODD) Characterizations Utilizing Subset Simulation. In Proceedings of the 2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC), 2024, pp. 1–10. <https://doi.org/10.1109/DASC62030.2024.10749586>.
46. Gu/egan, A. WG-114 AI Standards in Aviation.
47. International, S. G-34 Artificial Intelligence in Aviation. Available online: <https://standardsworks.sae.org/standards-committees/g-34-artificial-intelligence-aviation> (accessed on 2025-01-10).
48. G-34 Artificial Intelligence In Aviation Committee. Process Standard for Development and Certification/Approval of Aeronautical Safety-Related Products Implementing AI. Available online: <https://www.sae.org/standards/content/arp6983/> (accessed on 2024-11-16).
49. Kaakai, F.; Machrouh, J. Guide to the Preparation of Operational Concept Documents (ANSI/AIAA G-043B-2018), 2025.
50. Julian, K.D.; Kochenderfer, M.J. Guaranteeing Safety for Neural Network-Based Aircraft Collision Avoidance Systems. In Proceedings of the 2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC), San Diego, CA, USA, 09 2019; pp. 1–10. <https://doi.org/10.1109/dasc43569.2019.9081748>.
51. Christensen, J.M.; Zaeske, W.; Beck, J.; Friedrich, S.; Stefani, T.; Girija, A.A.; Hoemann, E.; Durak, U.; Köster, F.; Krüger, T.; et al. Towards Certifiable AI in Aviation: A Framework for Neural Network Assurance Using Advanced Visualization and Safety Nets. In Proceedings of the 2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC), San Diego, CA, USA, 09 2024; pp. 1–9. <https://doi.org/10.1109/dasc62030.2024.10749321>.
52. Damour, M.; De Grancey, F.; Gabreau, C.; Gauffriau, A.; Ginestet, J.B.; Hervieu, A.; Huraux, T.; Pagetti, C.; Ponsolle, L.; Clavière, A. Towards Certification of a Reduced Footprint ACAS-Xu System: A Hybrid ML-Based Solution. In Proceedings of the Computer Safety, Reliability, and Security, Cham, Switzerland, 2021; pp. 34–48. https://doi.org/10.1007/978-3-030-83903-1_3.
53. ISO. ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes – Requirements engineering. *ISO/IEC/IEEE 29148:2018(E)* 2018, pp. 1–104. <https://doi.org/10.1109/IEEESTD.2018.8559686>.
54. Verein zur Förderung der internationalen Standardisierung von Automatisierungs- und Meßsystemen (ASAM) e.V.. ASAM OpenODD Base Standard 1.0.0 Specification, 2025.
55. Mercedes-Benz Research & Development North America, Inc.. Introducing DRIVE PILOT: An Automated Driving System for the Highway. Available online: <https://group.mercedes-benz.com/dokumente/innovation/sonstiges/2023-03-06-vssa-mercedes-benz-drive-pilot.pdf> (accessed on 2024-11-07).
56. Shakeri, A. Formalization of Operational Domain and Operational Design Domain for Automated Vehicles. In Proceedings of the 2024 IEEE 24th International Conference on Software Quality, Reliability, and Security Companion (QRS-C), 04 2024, pp. 990–997. <https://doi.org/10.1109/QRS-C63300.2024.00131>.
57. Weiss, G.; Zeller, M.; Schoenhaar, H.; Fraunhofer, C.D.; Kreutz, A. Approach for Argumenting Safety on Basis of an Operational Design Domain. In Proceedings of the 2024 IEEE/ACM 3rd International Conference on AI Engineering – Software Engineering for AI (CAIN), 2024, pp. 184–193.
58. Adedjouma, M.; Botella, B.; Ibanez-Guzman, J.; Mantissa, K.; Proum, C.M.; Smaoui, A. Defining Operational Design Domain for Autonomous Systems: A Domain-Agnostic and Risk-Based Approach. In Proceedings of the SOSE 2024 - 19th Annual System of Systems Engineering Conference, Tacoma, WA, United States, 06 2024; pp. 166–171. <https://doi.org/10.1109/SOSE62659.2024.10620936>.

59. Lee, C.W.; Nayeer, N.; Garcia, D.E.; Agrawal, A.; Liu, B. Identifying the Operational Design Domain for an Automated Driving System through Assessed Risk. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), 2020, pp. 1317–1322. <https://doi.org/10.1109/IV47402.2020.9304552>.
60. Viola, N.; Corpino, S.; Fioriti, M.; Stesina, F.; et al. Functional analysis in systems engineering: Methodology and applications. In *Systems engineering-practice and theory*; InTech, 2012; pp. 71–96.
61. S-18 Aircraft and Sys Dev and Safety Assessment Committee. *Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment*, 1996. <https://doi.org/10.4271/ARP4761>.
62. Herrmann, M.; Witt, C.; Lake, L.; Guneshka, S.; Heinzemann, C.; Bonarens, F.; Feifel, P.; Funke, S. Using ontologies for dataset engineering in automotive AI applications. In Proceedings of the 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2022, pp. 526–531. <https://doi.org/10.23919/DATE54114.2022.9774675>.
63. Systems, M.P. General Properties and Characteristics of Sensors. Available online: <https://www.monolithicpower.com/en/learning/mpscholar/sensors/intro-to-sensors/general-properties-characteristics?srsId=AfmBOooPonoNMaH1XZkUdaUvY50wL1EeRCkqxWEdjWBG8xttduX4OQ1V> (accessed on 2024-12-10).
64. Dodge, S.; Karam, L. Understanding how image quality affects deep neural networks. In Proceedings of the 2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX), 2016, pp. 1–6. <https://doi.org/10.1109/QoMEX.2016.7498955>.
65. Jiang, L.; Wang, X. Dataset Construction through Ontology-Based Data Requirements Analysis. *Applied Sciences* **2024**, *14*. <https://doi.org/10.3390/app14062237>.
66. Paton, N.W.; Chen, J.; Wu, Z. Dataset Discovery and Exploration: A Survey. *ACM Computing Surveys* **2023**, *56*. <https://doi.org/10.1145/3626521>.
67. da Cunha Davison, J.a.C.; Tostes, P.I.; Guerra Carneiro, C.A. Framework Architecture for AI/ML Data Management for Safety-Critical Applications. In Proceedings of the 2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC), 2024, pp. 1–9. <https://doi.org/10.1109/DASC62030.2024.10748782>.
68. Kreuzberger, D.; Kühl, N.; Hirschl, S. Machine learning operations (mlops): Overview, definition, and architecture. *IEEE access* **2023**, *11*, 31866–31879.
69. Stanford Intelligent Systems Laboratory. VerticalCAS Repository, 2020.
70. Stanford Intelligent Systems Laboratory. HorizontalCAS Repository, 2020.
71. Christensen, J.M.; Anilkumar Girija, A.; Stefani, T.; Durak, U.; Hoemann, E.; Köster, F.; Krüger, T.; Hallerbach, S. Advancing the AI-Based Realization of ACAS X Towards Real-World Application, 2024. <https://doi.org/10.5281/ZENODO.13815668>.
72. RTCA, Inc.. Minimum Operational Performance Standards for Airborne Collision Avoidance System X (ACAS X) (ACAS Xa and ACAS Xo). techreport DO-385, GlobalSpec, Washington, DC, USA, 2018.
73. RTCA, Inc.. Minimum Operational Performance Standards for Airborne Collision Avoidance System Xu (ACAS Xu). techreport DO-386, GlobalSpec, Washington, DC, USA, 2020.
74. FlightGear developers & contributors. FlightGear. Available online: <https://www.flightgear.org/> (accessed on 2023-03-09).
75. EUROCAE. Minimum Aviation System Performance Standard for Detect and Avoid (Traffic) in Class A-C airspaces - with Corrigendum 1. Technical report, EUROCAE, Saint-Denis, France, 2022.
76. EUROCAE. ED-313. Technical report, EUROCAE, Saint-Denis, France, 2023.
77. Safety, S.A. GPWS. Available online: <https://skybrary.aero/gpws> (accessed on 2024-12-28).
78. Civil Aviation Authority of New Zealand. ADS-B in New Zealand. Available online: <https://www.nss.govt.nz/assets/nss/resources/2018-07-31-ADSB-FAQ-Document-V0.3.docx.pdf> (accessed on 2025-01-10).
79. Force, U.S.S. GPWS. Available online: <https://www.spaceforce.mil/About-Us/Fact-Sheets/Article/2197765/global-positioning-system/> (accessed on 2024-12-29).
80. Oceanic, N.; Administration, A. GPS Accuracy. Available online: <https://www.gps.gov/systems/gps/performance/accuracy/> (accessed on 2024-12-29).
81. Chryssanthacopoulos, J.P.; Kochenderfer, M.J. Collision avoidance system optimization with probabilistic pilot response models. In Proceedings of the Proceedings of the 2011 American Control Conference, 2011, pp. 2765–2770. <https://doi.org/10.1109/ACC.2011.5990776>.
82. Julian, K.D.; Lopez, J.; Brush, J.S.; Owen, M.P.; Kochenderfer, M.J. Policy compression for aircraft collision avoidance systems. In Proceedings of the 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), Sacramento, CA, USA, 09 2016; pp. 1–10. <https://doi.org/10.1109/DASC.2016.7778091>.

83. Julian, K.D.; Kochenderfer, M.J.; Owen, M.P. Deep Neural Network Compression for Aircraft Collision Avoidance Systems. *Journal of Guidance, Control, and Dynamics* **2019**, *42*, 598–608. <https://doi.org/10.2514/1.G003724>.
84. de Grancey, F.; Gerchinovitz, S.; Alecu, L.; Bonnin, H.; Dalmau, J.; Delmas, K.; Mamalet, F. On the Feasibility of EASA Learning Assurance Objectives for Machine Learning Components. In Proceedings of the ERTS 2024, Toulouse, France, 06 2024; ERTS2024. Best paper award at ERTS 2024.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.